



WHITE PAPER: CONSTRAINT-BASED SYSTEMS

by Ronald M. Shapiro

What is a Constraint-Based System?

Constraint-Based systems use the inherent constraints in a problem to rule out impossible alternatives before and during the search for a solution. Constraints can be just about any relationship between elements of a problem, such as "a shipping container must be larger than its contents" or "an employee cannot work consecutive eight hour shifts". Registering or "posting" constraints limits the possible values for the elements. The remaining values are manipulated to find one or all possible solutions. Systems typically search for the single solution which meets a goal, such as minimizing shipping costs, maximizing resource utilization, or minimizing production time.

Why haven't I heard of this technology?

Constraint-Based Reasoning was invented in the US in the early 1970's as a way to enhance logic programming, in particular PROLOG. The US commercial AI industry largely ignored the technology as it evolved in Europe. With the introduction of constraint-Based development products written in C++, the technology is ready to cross the Atlantic again. Churchill Systems Inc. recently sent a representative to the UK to evaluate the technology.

When are Constraint-Based Systems used?

Constraint Solving Systems are extremely well suited to the following applications:

- Scheduling
- Planning
- Bin Packing
- Resource Allocation
- Routing

There are two major motivations for using this technology. First, the declarative style of the problem representation offers development and maintenance advantages. Second, Constraint-Based Systems are extremely efficient because they thrive on restrictions; the more constraints on the solution, the faster a solution can be found. In comparison, conventional systems slow down as additional complexity is introduced into the problem.

How do Constraint-Based Systems compare to other AI techniques?

Most Artificial Intelligence (AI) problems involve searching. For example, Expert Systems either search the problem space from a known outcome to the causes of that outcome (known as 'backward chaining' or 'goal driven') or from known causes to possible outcomes (called 'forward chaining' or 'data driven'). At the extreme end of inefficiency is a technique known as exhaustive search, which simply involves trying all combinations of possible causes to see if any of the combinations produce the desired result. Constraints are useful because they eliminate inconsistent values from consideration before the search routines begin. Also, as new values are tried during the search the constraints are re-evaluated to further eliminate impossible alternatives. This process of evaluating the constraints and selecting ever narrowing alternatives continues until a solution is found, or one or more of the constraints cannot be satisfied.



What are the steps involved in designing an application?

There are three major tasks involved when using a Constraint-Based System. The first task is to model in the software system the components of the problem. Suppose we have a box in which we wish to place products for shipping. This box has a maximum weight capacity of 10 lbs. The product line consists of two items: widget A has possible weights of between 1 and 10 lbs., and widget B has possible weights of between 5 and 15 lbs. The goal is to pack the box with the two items to achieve the highest weight possible. From this description the model consists of:

- a packing box with attribute weight with values 0 through 10
- widgets with attributes name and weight with values 1 through 15

Also at this stage the quantities to be searched for and optimized are determined. The unknown value to be searched for (and maximized) in this case is the total weight of the box.

The second step is to define the constraints which apply to the components of the problem. This step typically involves the personnel who are considered experts at solving the problem currently (called 'domain experts'), and a knowledge engineer who is skilled in translating these constraints into a format suitable for the computer. To continue our example the constraints are:

- the sum of the weights of widget A and widget B must be 10 lbs. or less
- the sum of the items must be two
- one each of widget A and widget B must be used

By applying these constraints it is immediately known that:

- widget A must have a weight between 1 and 5, since the smallest combined weight would be greater than 10 lbs otherwise.
- widget B must have a weight between 5 and 9 lbs
- widget A must be in the box
- widget B must be in the box

As is typical in Constraint-Based Systems, the application of the constraints alone is not sufficient to find a solution to the problem. The third step in designing the application is to define the search method used to find one or all solutions to the problem. This task is also performed with the domain expert and the knowledge engineer working together. The search methods used can come from operations research, company rules, or brute force. A rule of thumb is to start searching on the element which has the smallest number of options, since the chances of selecting the correct one are higher. In this case, the search could be started by assigning the largest value to widget A, since we know that we want the maximum total weight possible. By setting the weight of widget A to 5 lbs. and re-evaluating the constraints the only value for the weight of widget B is 5 lbs. This is then the solution to the problem.

Note that if we did not start selecting weight values starting with the maximum, we would have had to perform several more iterations until a maximal solution could be found. Even so, by applying the initial constraints many impossible combinations were avoided.

What is the conclusion?

Constraint-Based Systems are a rapidly emerging technology that can provide a great benefit in solving complex problems in such areas as scheduling, planning, and resource allocation. The major motivations in using this type of problem solving method are the ease of expressing the problem (which has both short and long term benefit) and the efficient computer performance in finding solutions to the problem.